storing an object in the first memory device; and

determining that the stored object is not accessible based upon lack of a corresponding entry in the directory table.

71. (NEW) A method as in claim 55 further comprising:

A13 cncld

retrieving a request for an object stored in the first memory device;

hashing a file specifier associated with the requested object to locate an entry in the directory table; and

comparing the file specifier associated with the requested object to data information stored along with the requested object in the first memory device to identify whether a file is the requested object.

72. (NEW) A method as in claim 71, wherein the file specifier is a URL (Uniform Resource Locator).

REMARKS

Claims 1-12 were previously submitted in the present application. Claims 13-72 are being added by way of this amendment. Thus, following entry of this amendment, claims 1-72 will be pending.

The specification is being amended to correct self-evident and typographical errors. No new matter is being added to the specification.

CONCLUSION

It is submitted that all claims are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned Attorney at (978) 341-0036.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By Paul P. Kriz

Paul P. Kriz
Registration No. 45,752
Telephone: (978) 341-0036
Facsimile: (978) 341-0136

Concord, MA 01742-9133

Dated:
October 21, 2002

MARKED UP VERSION OF AMENDMENTS

Specification Amendments Under 37 C.F.R. § 1.121(b)(1)(iii)

Replace the paragraph at page 3, lines 17 through 24 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph:

The present invention seeks to alleviate these and other difficulties by providing a non-hierarchical or linear directory structure for a mass storage unit such as a disk. The directory structure can be kept in an auxiliary semiconductor memory. The disk is partitioned into segments of equal size. The directory structure presumes that data objects reside wholly and contiguously within a given area of the disk segments. While a variable number of objects may be stored within each segment, a given object is not allowed to occupy more than one segment. During a storage operation, objects are assigned to successive segments in a round-robin fashion, to equalize segment utilization.

Please replace the paragraphs at page 5, line 20 through page 6, line 18 with the below paragraphs marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraphs:

In the illustrated network, both the Web home servers 20 and the Web cache servers 16 provide for storage for documents or other data objects in the form of mass storage devices 18 and 21. Cache servers 16 are associated with cache mass storage 18 which may include disk storage 18-1[-1] and memory storage [18-1-2] 184-2. Similarly, the home server 20 has associated disk storage 21-1-1 and memory storage 21-1-2.

It is assumed in the following discussion that the network 10 is the Internet, that information is stored at the servers 16 and 20 [is the] in the form of Web data objects such as hypertext transfer protocol (HTTP) documents and that request messages are routed among the computers in order to

fetch the documents by references such as in the form of uniform resource locator (URL) using standard network protocol such as the transmission control protocol/Internet protocol (TCP/IP). However, this one example is described with the understanding that many other types of computer architectures, network architectures, and other types of data objects and/or protocols may make advantageous use of the teachings of the present invention.

In any event, in such a network 10, document requests originate typically at one of the client computers such as client 12-1 in the form of [URLs] a URL's specification for an HTTP document stored at a home server 20. The message is formulated as a request by the client in the HTTP protocol for the [homer] home server 20 to return a copy of a data object that is presently stored at the home server 20, such as a file stored on the disk [21-1] 21-1-1. The document request is passed through one or more routers 14, such as routers 14-1, 14-2, 14-3, in the direction of the illustrated arrows, on its way towards the home server 20. The request may be intercepted at any of the intermediate nodes that have a cache server 16 [are] associated with them. Cache servers 16 intercept document requests and determine if the requested data object can be served from one of its local disks 18. In particular, if the requested data object is stored on one of the disks 18 associated with one of the cache servers 16, the requested content will instead be delivered from the cache server 18 rather than the home server 20.

Please replace the paragraph at page 9, lines 20 through 28 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph:

Turning attention briefly to Fig. 5, it is seen [at] that each data partition 200 can be considered to be a type of ring structure. The ring 210 thus can be thought of as consisting of a number of segments 220-0, 220-1, ..., 220-n-1 where n is the total number of segments. In the indicated preferred embodiment, there are 128 segments as previously described. An exemplary segment, such as the segment 220-2 illustrated, may contain any number of data objects. In the illustrated example, segment 220-2 contains five data objects 230 (labeled object 1, object 2, ..., object 5). Another portion of the segment 220-2 does not yet contain any objects 230 and therefore can be considered to be free space 240.

Please replace the paragraph at page 10, lines 4 through 11 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph:

It should also be recognized that the number of active segments, such as one segment 220 per ring 210, are determined and that objects are assigned to successive active segments in a round-robin fashion. For an empty ring 210, the active segment 220 is the first empty segment in a ring. Once a segment 220 is full, the next empty segment is selected to become an active segment. If no empty segments 220 are found, the ring is full. Whenever data is written into a segment, a segment time stamp may be updated to the present time. For more information in connection with the stores of segments 220 in a partition 200 will be discussed in connection with Figs. 5 and 6 and others below.

Please replace the paragraph at page 11, lines 20 through 22 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph:

The lower M bits of a hash value are thus used to specify a particular directory entry 310 within a certain directory block 300. Each directory block 300 thus consists of a number of directory entries 310, as determined by the lower hash value M.

Please replace the paragraph at page 14, lines 3 through 5 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph:

If no empty segments 220 are available, that partition 200 is declared as full until garbage collection is available to clear out a full segment 220 to [market] mark it as empty. The garbage collection process is described in further detail below.

Please replace the paragraph at page 16, lines 6 through 11 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph:

An "expired data" garbage collection process is similar to the oldest data method. However, when segments are scanned, the event handler attempts to find a segment whose data has already expired. If one cannot be found available, meaning [at] that all segments 220 hold unexpired data, then the event handler will simply reschedule itself to run when the first segment 220 is due to expire. The time can be determined by the expiration date field 466.

Please replace the paragraph at page 18, lines 15 through 20 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph:

The memory object 500 holds a working set of one or more contiguous buffers 520-0, ... 520-3 which are used to hold the in-memory version of the on-disk data object 230. The sizes of the buffers 520 for each memory object 500 is the same as the size of the on-disk object up to a configurable maximum. For example, this might be [65] 64 kilobytes. All requests to either read or write objects larger than this maximum buffer size must then, therefore, share buffers for that object.

Claim Amendments Under 37 C.F.R. §1.121(c)(1)(ii)

4. (Amended) A file subsystem as in claim 1 wherein the file subsystem is incorporated in one of [fo] a Web page home server or Web page cache server.

6. (Amended) A file subsystem as in claim 5 wherein the directory entry specifies at least a partition [segment], starting block number, and size for the data object.

12. (Amended) A storage subsystem as in claim 1 wherein a new data object is written to overwrite a previously stored [an oldest] data object in a segment [when a segment is full] based on a time stamp indicating when the previous data object was stored.